**Baza de date a unui lant de restaurante**

Realizat de: Petchi Andrei Luca Grupa 234

1. **Prezentati pe scurt baza de date ( utilitatea ei ).**

Pentru acest proiect , am ales realizarea bazei de date a unui lant de restaurante cu specific japonez , format din ,,Japanos Fusion’’, ,,Japanos By The Lake ”, ,, Japanos Tineretului ”, ,, Japanos Takeaway ” si ,,Japanos Dorobanti”. Restaurantele functioneaza pe un sistem clasic, clientului ii este preluata comanda de catre un ospatar, urmand ca aceasta sa fie preparata de un bucatar. Acesta plateste la final nota, avand posibilitatea platii cash sau card. Clientii au posibilitatea de a face rezervari sau comenzi online. In cazul comenziilor online , comanda este livrata de catre un curier din cadrul restaurantului solicitat . Baza de date propusa faciliteaza managementul restaurantelor din retea, angajatii , clientii si comenziile fiind reprezentati de entitatile “ANGAJATI” , “CLIENTI” respectiv “COMANDA” tinand evidenta rolului fiecarui angajat, performanta lor si datele fiecarei comenzi. De asemenea , baza de date ofera informatii despre stocuri, vanzari cat si preferintele clientilor, ajuta la luarea deciziilor informate, optimizand costurile, cresterea eficientei operationale si imbunatatirea experientei clientilor.

**Reguli**

-Fiecare angajat lucreaza intr-o singura locatie, iar fiecare locatie la randul ei are cel putin un angajat pentru fiecare departament.

-Rezervarea unei mese este posibila cat si comandarea prin livrare.

-Angajatii sunt impartiti in :chefi,staff si curieri, fiecare angajat avand id-ul unic de angajat dar si id-ul pentru departamentul din care fac parte.

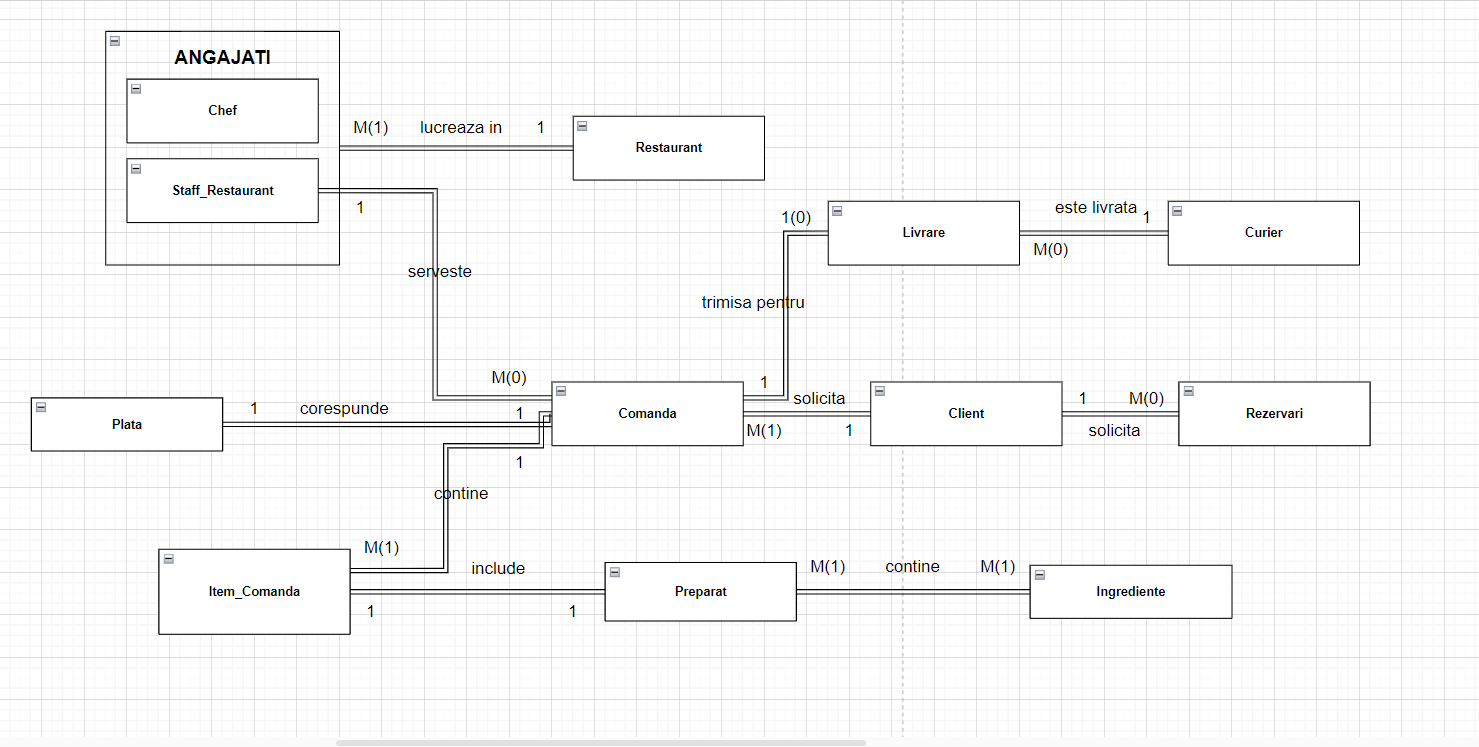
-Fiecare comanda este unica,identifiabila prin id. -Orice comanda are nota de plata.

-O comanda trimisa prin livrare este preluata de un curier. -Meniul contine diferite preparate,pretul lor fiind flexibil in functie de perioada.

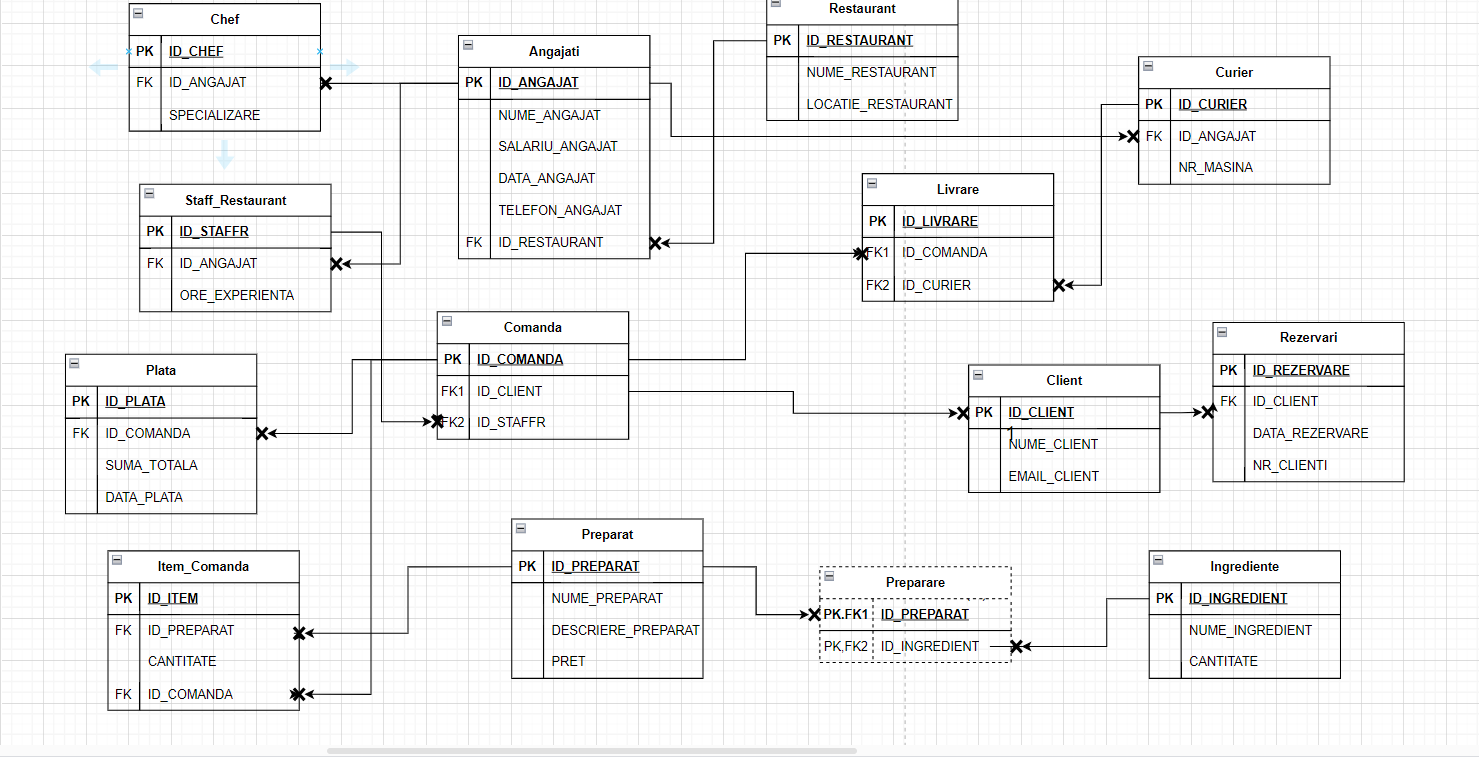
-Angajatii au salariul in functie de vechimea lor.

-Fiecare curier are o masina pentru livrare, costurile ei fiind decontate de catre restaurant.

**2. Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română.**



**3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.**



**4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).**

create TABLE RESTAURANT(

ID\_RESTAURANT int not null primary key,

NUME\_RESTAURANT varchar2(50),

LOCATIE\_RESTAURANT varchar2(50)

);

create TABLE ANGAJATI(

ID\_ANGAJAT int not null primary key,

NUME\_ANGAJAT varchar2(50),

SALARIU\_ANGAJAT int ,

DATA\_ANGAJAT date,

TELEFON\_ANGAJAT varchar2(10),

ID\_RESTAURANT not null,

foreign key(ID\_RESTAURANT) references RESTAURANT(ID\_RESTAURANT)

);

create TABLE STAFF\_RESTAURANT(

ID\_STAFFR int not null primary key,

ID\_ANGAJAT not null,

LUNI\_EXPERIENTA int,

foreign key(ID\_ANGAJAT) references ANGAJATI(ID\_ANGAJAT)

);

create TABLE CHEF(

ID\_CHEF int not null primary key,

ID\_ANGAJAT not null,

SPECIALIZARE varchar2(20),

foreign key(ID\_ANGAJAT) references ANGAJATI(ID\_ANGAJAT)

);

create TABLE CLIENTI(

ID\_CLIENT int not null primary key,

NUME\_CLIENT varchar2(50),

EMAIL\_CLIENT varchar2(50)

);

create TABLE REZERVARI(

ID\_REZERVARE int not null primary key,

ID\_CLIENT not null,

DATA\_REZERVARE date,

NR\_CLIENTI int,

foreign key(ID\_CLIENT) references CLIENT(ID\_CLIENT)

);

create TABLE COMANDA(

ID\_COMANDA int not null primary key,

ID\_CLIENT not null,

ID\_STAFFR not null,

foreign key(ID\_CLIENT) references CLIENT(ID\_CLIENT),

foreign key(ID\_STAFFR) references STAFF\_RESTAURANT(ID\_STAFFR)

);

create TABLE PREPARAT(

ID\_PREPARAT int not null primary key,

NUME\_PREPARAT varchar2(20),

DESCRIERE\_PREPARAT varchar2(200)

PRET int,

);

create TABLE INGREDIENTE(

ID\_INGREDIENT int not null primary key,

NUME\_INGREDIENT varchar2(50),

Cantitate int,

foreign key(ID\_COMANDA) references COMANDA(ID\_COMANDA)

);

create TABLE PLATA(

ID\_PLATA int not null primary key,

ID\_COMANDA not null,

SUMA\_TOTALA decimal,

DATA\_PLATA date,

foreign key(ID\_COMANDA)references COMANDA(ID\_COMANDA)

);

create TABLE CURIER(

ID\_CURIER int not null primary key,

ID\_ANGAJAT not null,

NR\_MASINA int,

foreign key(ID\_ANGAJAT)references ANGAJATI(ID\_ANGAJAT)

);

create TABLE LIVRARE(

ID\_LIVRARE int not null primary key,

ID\_COMANDA not null,

ID\_CURIER not null,

foreign key(ID\_COMANDA) references COMANDA(ID\_COMANDA),

foreign key(ID\_CURIER) references CURIER(ID\_CURIER)

);

create TABLE ITEM\_COMANDA(

ID\_ITEM int not null primary key,

CANTITATE int,

ID\_PREPARAT not null,

ID\_COMANDA not null,

foreign key(ID\_PREPARAT) references PREPARAT(ID\_PREPARAT),

foreign key(ID\_COMANDA) references COMANDA(ID\_COMANDA)

);

Create TABLE PREPARARE(

ID\_PREPARAT not null,

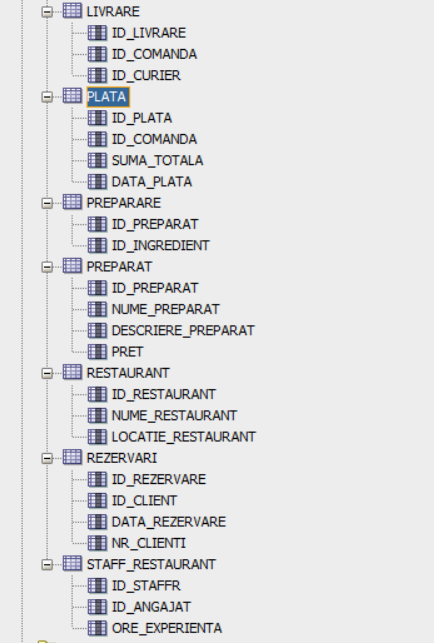
ID\_INGREDIENT not null,

Foreign key(ID\_PREPARAT) references PREPARAT(ID\_PREPARAT),

Foreign key(ID\_INGREDIENT) references INGREDIENT(ID\_INGREDIENT)

);





**5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).**

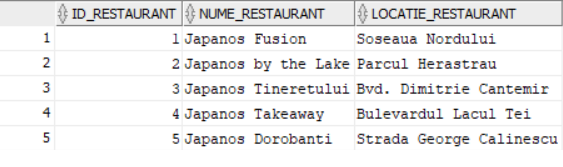
insert into RESTAURANT VALUES(1,'Japanos Fusion','Soseaua Nordului');

insert into RESTAURANT VALUES(2,'Japanos by the Lake','Parcul Herastrau');

insert into RESTAURANT VALUES(3,'Japanos Tineretului','Bvd. Dimitrie Cantemir');

insert into RESTAURANT VALUES(4,'Japanos Takeaway','Bulevardul Lacul Tei');

insert into RESTAURANT VALUES(5,'Japanos Dorobanti','Strada George Calinescu');



SELECT \* FROM ANGAJATI;

insert into ANGAJATI VALUES(111,'Popescu Radu',4400,to\_date('01/03/2022','dd/mm/yyyy'),'0721020282',1);

insert into ANGAJATI VALUES(112,'Antonescu Mihai',5000,to\_date('02/05/2021','dd/mm/yyyy'),'0724040502',2);

insert into ANGAJATI VALUES(113,'Andrei Stefan Catalin',5500,to\_date('10/07/2020','dd/mm/yyyy'),'0732028213',3);

insert into ANGAJATI VALUES(114,'Nastea Calin',4400,to\_date('01/10/2022','dd/mm/yyyy'),'0743209876',4);

insert into ANGAJATI VALUES(115,'Georgescu Maria',4400,to\_date('25/05/2022','dd/mm/yyyy'),'0721384821',5);

insert into ANGAJATI VALUES(116,'Ionescu Ana',6000,to\_date('14/03/2017','dd/mm/yyyy'),'0755338212',1);

insert into ANGAJATI VALUES(117,'Nedelcu Ana',5000,to\_date('20/04/2020','dd/mm/yyyy'),'0722308202',2);

insert into ANGAJATI VALUES(118,'Ionita Alexandru',5000,to\_date('14/12/2021','dd/mm/yyyy'),'0723458922',3);

insert into ANGAJATI VALUES(119,'Oprea Laurentiu',5000,to\_date('01/06/2021','dd/mm/yyyy'),'0744994421',4);

insert into ANGAJATI VALUES(120,'Voinea Alexandru',6000,to\_date('12/03/2019','dd/mm/yyyy'),'0721982394',5);

insert into ANGAJATI VALUES(121,'Ceausu Florina',7500,to\_date('04/06/2016','dd/mm/yyyy'),'0721992182',1);

insert into ANGAJATI VALUES(122,'Velcea Maria',5500,to\_date('05/02/2020','dd/mm/yyyy'),'0723402345',1);

insert into ANGAJATI VALUES(123,'Ammari Abdul',7500,to\_date('06/10/2015','dd/mm/yyyy'),'0723449942',2);

insert into ANGAJATI VALUES(124,'Dumitru Tudor',4400,to\_date('20/09/2022','dd/mm/yyyy'),'0728992314',3);

insert into ANGAJATI VALUES(125,'Spataru Vlad',5000,to\_date('10/07/2021','dd/mm/yyyy'),'0788991234',4);

insert into ANGAJATI VALUES(126,'Dumitru Vlad',5000,to\_date('15/05/2021','dd/mm/yyyy'),'0799124422',5);

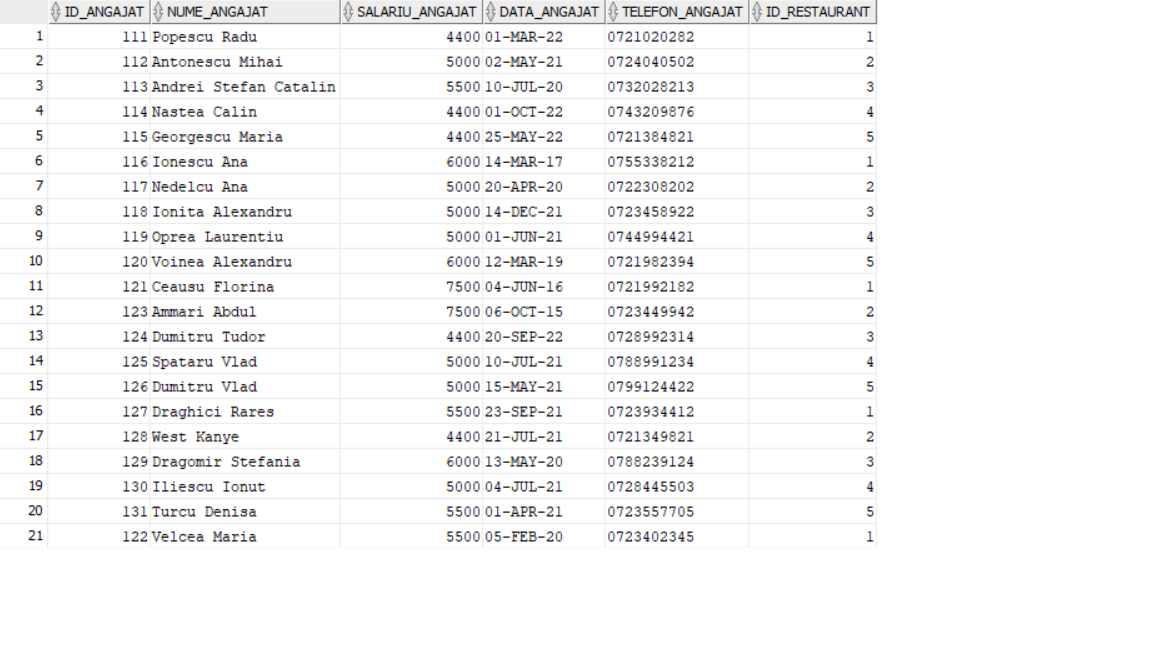
insert into ANGAJATI VALUES(127,'Draghici Rares',5500,to\_date('23/09/2021','dd/mm/yyyy'),'0723934412',1);

insert into ANGAJATI VALUES(128,'West Kanye',4400,to\_date('21/07/2021','dd/mm/yyyy'),'0721349821',2);

insert into ANGAJATI VALUES(129,'Dragomir Stefania',6000,to\_date('13/05/2020','dd/mm/yyyy'),'0788239124',3);

insert into ANGAJATI VALUES(130,'Iliescu Ionut',5000,to\_date('04/07/2021','dd/mm/yyyy'),'0728445503',4);

insert into ANGAJATI VALUES(131,'Turcu Denisa',5500,to\_date('01/04/2021','dd/mm/yyyy'),'0723557705',5);



SELECT \* FROM CLIENT;

insert into CLIENT(ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Ion Dan','iondan@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Oprea Ion','oprion@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Letea Radu','letear@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Dobrin Letitia','dobr@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Camelia Carmen','carmeliac@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Rosoiu Luca','lucarosoiu@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Florea Andrei','floreandr@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Pintilie Ana','anapintilie@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Mihail Alexandra','alexandramihail@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Omar Ismail ','ismailomul@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Bisoc Alina','alinab@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Pene Florin','peneflorin@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Nitu Miruna','mirunadent@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Popovici David','davidpopovici@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Anghel Luminita','luminitaanghel@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Traistariu Mihai','mihaitraistariu@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Dumitrescu Bogdan','bogdand@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Dumitrascu Maria','mariadum@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Pavel Alina','pavelalina@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Ionescu Roxana','roxanaione@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Ionescu Mihai','ionescumihai@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Panait Razvan','panaitrazvan@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

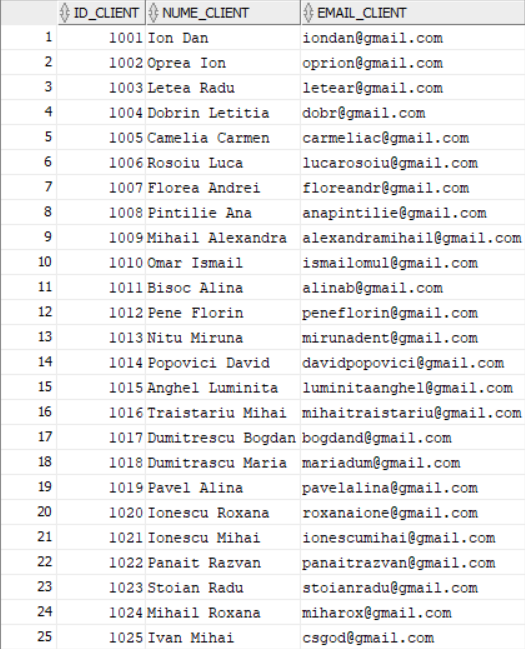
VALUES(seq\_id\_client.NEXTVAL,'Stoian Radu','stoianradu@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Mihail Roxana','miharox@gmail.com');

insert into CLIENT (ID\_CLIENT,NUME\_CLIENT,EMAIL\_CLIENT)

VALUES(seq\_id\_client.NEXTVAL,'Ivan Mihai','csgod@gmail.com');



SELECT \* FROM STAFF\_RESTAURANT;

insert into STAFF\_RESTAURANT VALUES(501,114,7);

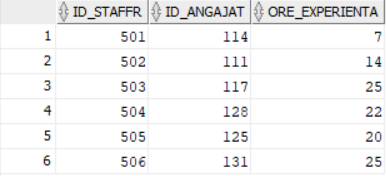
insert into STAFF\_RESTAURANT VALUES(502,111,14);

insert into STAFF\_RESTAURANT VALUES(503,117,25);

insert into STAFF\_RESTAURANT VALUES(504,128,22);

insert into STAFF\_RESTAURANT VALUES(505,125,20);

insert into STAFF\_RESTAURANT VALUES(506,131,25);



SELECT \* FROM COMANDA;

insert into COMANDA VALUES(10,1004,501);

insert into COMANDA VALUES(11,1001,501);

insert into COMANDA VALUES(12,1010,502);

insert into COMANDA VALUES(13,1012,502);

insert into COMANDA VALUES(14,1014,503);

insert into COMANDA VALUES(15,1013,503);

insert into COMANDA VALUES(16,1001,503);

insert into COMANDA VALUES(17,1009,503);

insert into COMANDA VALUES(18,1008,503);

insert into COMANDA VALUES(19,1006,504);

insert into COMANDA VALUES(20,1013,504);

insert into COMANDA VALUES(21,1015,505);

insert into COMANDA VALUES(22,1020,506);

insert into COMANDA VALUES(23,1021,503);

insert into COMANDA VALUES(24,1022,505);

insert into COMANDA VALUES(25,1023,503);

insert into COMANDA VALUES(26,1024,501);

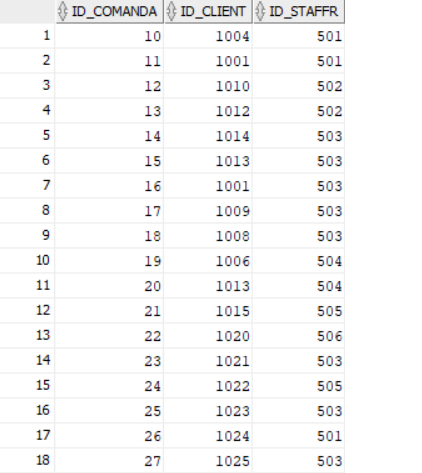
insert into COMANDA VALUES(27,1025,503);

insert into COMANDA VALUES(28,1001,501);

insert into COMANDA VALUES(29,1001,505);

insert into COMANDA VALUES(30,1001,505);

insert into COMANDA VALUES(31,1001,506);



SELECT \* FROM CHEF;

insert into CHEF VALUES(701,120,'Sushi');

insert into CHEF VALUES(702,121,'Sushi');

insert into CHEF VALUES(703,123,'Sushi');

insert into CHEF VALUES(704,116,'Ramen');

insert into CHEF VALUES(705,112,'Ramen');

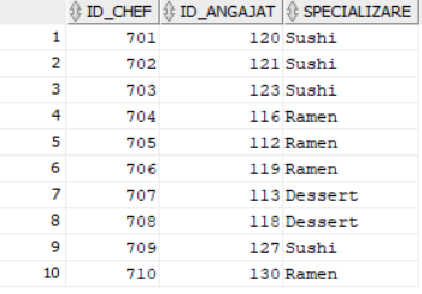
insert into CHEF VALUES(706,119,'Ramen');

insert into CHEF VALUES(707,113,'Dessert');

insert into CHEF VALUES(708,118,'Dessert');

insert into CHEF VALUES(709,127,'Sushi');

insert into CHEF VALUES(710,130,'Ramen');



SELECT \* FROM INGREDIENTE;

insert into INGREDIENTE VALUES(801,'Ton',90);

insert into INGREDIENTE VALUES(802,'Somon',55);

insert into INGREDIENTE VALUES(803,'Orez',231);

insert into INGREDIENTE VALUES(804,'Sos Soia',30);

insert into INGREDIENTE VALUES(805,'Creveti',40);

insert into INGREDIENTE VALUES(806,'Wasabi',70);

insert into INGREDIENTE VALUES(807,'Crab',220);

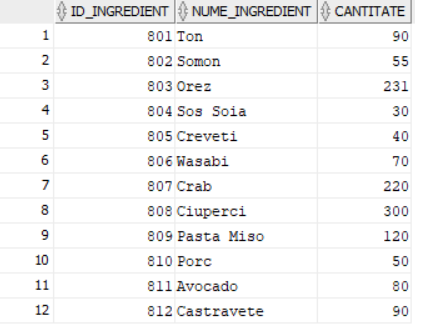
insert into INGREDIENTE VALUES(808,'Ciuperci',300);

insert into INGREDIENTE VALUES(809,'Pasta Miso',120);

insert into INGREDIENTE VALUES(810,'Porc',50);

insert into INGREDIENTE VALUES(811,'Avocado',80);

insert into INGREDIENTE VALUES(812,'Castravete',90);



SELECT \* FROM PREPARAT;

insert into PREPARAT VALUES(901,'Nigiri Ton','File de ton pe un pat de orez presat ',801);

insert into PREPARAT VALUES(902,'Nigiri Somon','File de somon pe un pat de orez presat ',802);

insert into PREPARAT VALUES(903,'Maki Ton','Rulou Maki clasic cu ton',801);

insert into PREPARAT VALUES(904,'Maki Somon','Rulou Maki clasic cu somon',802);

insert into PREPARAT VALUES(905,'Maki Avocado','Rulou Maki clasic cu avocado',811);

insert into PREPARAT VALUES(906,'Maki Spicy Somon','Rulou Maki clasic somon cu sosul spicy Japanos',802);

insert into PREPARAT VALUES(907,'Maki California','Rulou Maki care contine crab,avocado , castravete si seminte de susan',807);

insert into PREPARAT VALUES(908,'Maki Philadelphia','Rulou Maki care contine somon, cream cheese si castravete',802);

insert into PREPARAT VALUES(909,'Ramen Veggie','Ramen vegetarian pe baza de ciuperci shiitake',808);

insert into PREPARAT VALUES(910,'Ramen Porc','Ramenul clasic, construit in jurul porcului in stil chashu ',810);

insert into PREPARAT VALUES(911,'Supa Miso','Supa traditionala japoneza, ingredientul principal fiind pasta miso',809);

insert into PREPARAT VALUES(912,'Mochi','Prajitura japoneza facuta din orez',803);

UPDATE PREPARAT SET PRET = 20 WHERE ID\_PREPARAT=901;

UPDATE PREPARAT SET PRET = 21 WHERE ID\_PREPARAT=902;

UPDATE PREPARAT SET PRET = 25 WHERE ID\_PREPARAT=903;

UPDATE PREPARAT SET PRET = 30 WHERE ID\_PREPARAT=904;

UPDATE PREPARAT SET PRET = 20 WHERE ID\_PREPARAT=905;

UPDATE PREPARAT SET PRET = 30 WHERE ID\_PREPARAT=906;

UPDATE PREPARAT SET PRET = 30 WHERE ID\_PREPARAT=907;

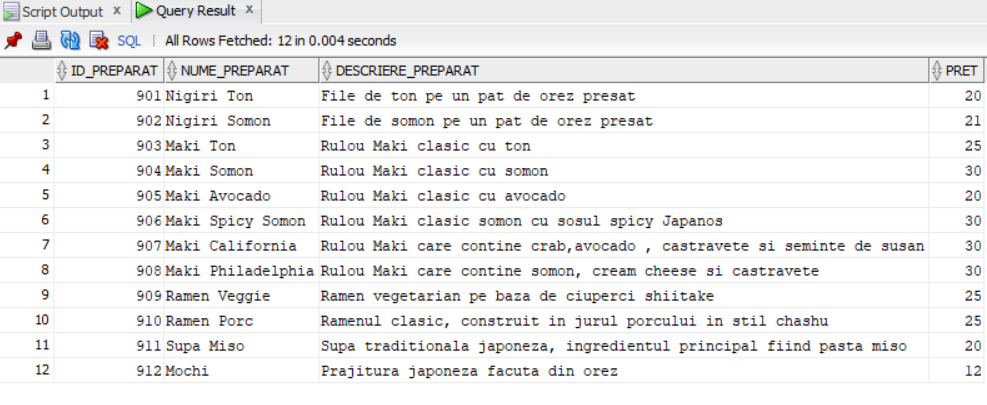
UPDATE PREPARAT SET PRET = 30 WHERE ID\_PREPARAT=908;

UPDATE PREPARAT SET PRET = 25 WHERE ID\_PREPARAT=909;

UPDATE PREPARAT SET PRET = 25 WHERE ID\_PREPARAT=910;

UPDATE PREPARAT SET PRET = 20 WHERE ID\_PREPARAT=911;

UPDATE PREPARAT SET PRET = 12 WHERE ID\_PREPARAT=912;



SELECT \* FROM CURIER;

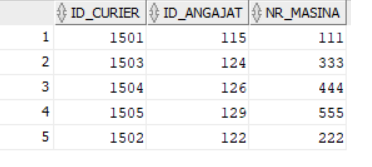
insert into CURIER VALUES(1501,115,111);

insert into CURIER VALUES(1502,122,222);

insert into CURIER VALUES(1503,124,333);

insert into CURIER VALUES(1504,126,444);

insert into CURIER VALUES(1505,129,555);



SELECT \* FROM ITEM\_COMANDA;

insert into ITEM\_COMANDA VALUES(651,1,901,10);

insert into ITEM\_COMANDA VALUES(652,2,905,11);

insert into ITEM\_COMANDA VALUES(653,4,903,12);

insert into ITEM\_COMANDA VALUES(654,3,906,13);

insert into ITEM\_COMANDA VALUES(655,1,907,14);

insert into ITEM\_COMANDA VALUES(656,1,910,15);

insert into ITEM\_COMANDA VALUES(657,2,910,16);

insert into ITEM\_COMANDA VALUES(658,2,903,17);

insert into ITEM\_COMANDA VALUES(659,1,901,18);

insert into ITEM\_COMANDA VALUES(660,3,902,19);

insert into ITEM\_COMANDA VALUES(661,4,904,20);

insert into ITEM\_COMANDA VALUES(662,5,905,21);

insert into ITEM\_COMANDA VALUES(663,6,912,22);

insert into ITEM\_COMANDA VALUES(664,2,911,13);

insert into ITEM\_COMANDA VALUES(665,3,906,14);

insert into ITEM\_COMANDA VALUES(666,4,908,13);

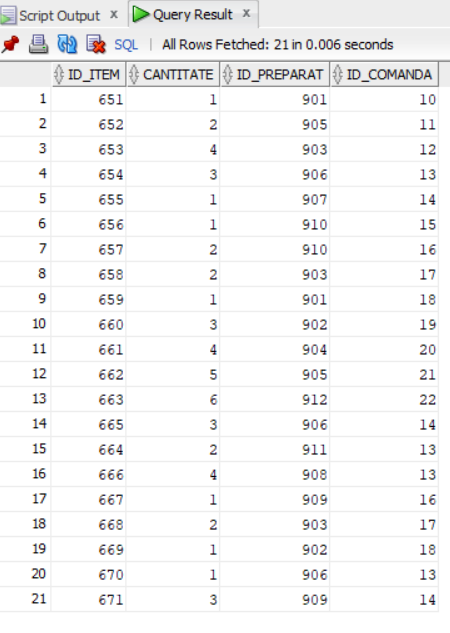
insert into ITEM\_COMANDA VALUES(667,1,909,16);

insert into ITEM\_COMANDA VALUES(668,2,903,17);

insert into ITEM\_COMANDA VALUES(669,1,902,18);

insert into ITEM\_COMANDA VALUES(670,1,906,13);

insert into ITEM\_COMANDA VALUES(671,3,909,14);



SELECT \* FROM PLATA;

insert into PLATA VALUES(350,10,48,to\_date('10/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(351,11,50,to\_date('08/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(352,12,80,to\_date('19/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(353,13,33,to\_date('18/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(354,14,166,to\_date('16/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(355,15,35,to\_date('18/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(356,16,61,to\_date('19/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(357,17,118,to\_date('18/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(358,18,141,to\_date('16/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(359,19,25,to\_date('17/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(360,20,60,to\_date('20/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(361,21,32,to\_date('21/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(362,22,20,to\_date('22/05/2023','dd/mm/yyyy'));

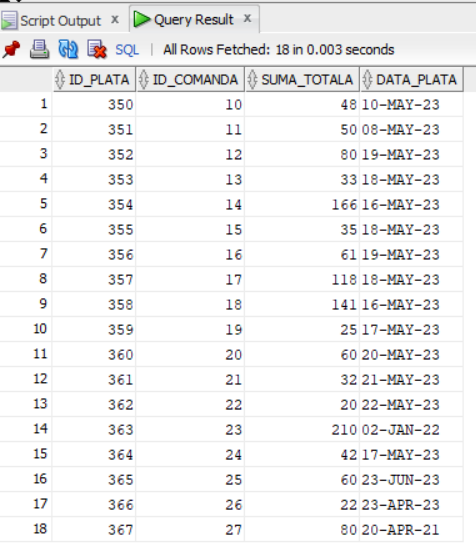
insert into PLATA VALUES(363,23,210,to\_date('02/01/2022','dd/mm/yyyy'));

insert into PLATA VALUES(364,24,42,to\_date('17/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(365,25,60,to\_date('23/06/2023','dd/mm/yyyy'));

insert into PLATA VALUES(366,26,22,to\_date('23/04/2023','dd/mm/yyyy'));

insert into PLATA VALUES(367,27,80,to\_date('20/04/2021','dd/mm/yyyy'));



SELECT \* FROM REZERVARI;

insert into REZERVARI VALUES(6001,1001,to\_date('10/05/2023','dd/mm/yyyy'),3);

insert into REZERVARI VALUES(6002,1002,to\_date('08/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6003,1003,to\_date('19/05/2023','dd/mm/yyyy'),4);

insert into REZERVARI VALUES(6004,1004,to\_date('18/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6005,1005,to\_date('16/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6006,1006,to\_date('18/05/2023','dd/mm/yyyy'),2);

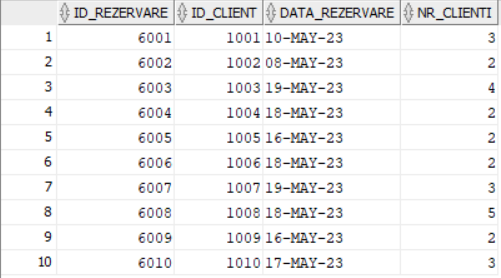
insert into REZERVARI VALUES(6007,1007,to\_date('19/05/2023','dd/mm/yyyy'),3);

insert into REZERVARI VALUES(6008,1008,to\_date('18/05/2023','dd/mm/yyyy'),5);

insert into REZERVARI VALUES(6009,1009,to\_date('16/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6010,1010,to\_date('17/05/2023','dd/mm/yyyy'),3);

insert into REZERVARI VALUES(6011,1020,to\_date('17/05/2023','dd/mm/yyyy'),3);



SELECT \* FROM LIVRARE;

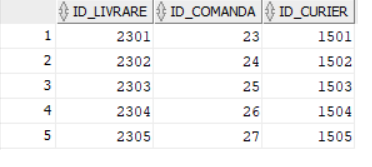
insert into LIVRARE VALUES(2301,23,1501);

insert into LIVRARE VALUES(2302,24,1502);

insert into LIVRARE VALUES(2303,25,1503);

insert into LIVRARE VALUES(2304,26,1504);

insert into LIVRARE VALUES(2305,27,1505);



SELECT \* FROM PREPARARE;

Insert into PREPARARE VALUES(901,802);

Insert into PREPARARE VALUES(902,809);

Insert into PREPARARE VALUES(903,808);

Insert into PREPARARE VALUES(904,804);

Insert into PREPARARE VALUES(905,801);

Insert into PREPARARE VALUES(906,801);

Insert into PREPARARE VALUES(907,803);

Insert into PREPARARE VALUES(908,803);

Insert into PREPARARE VALUES(909,806);

Insert into PREPARARE VALUES(910,807);

Insert into PREPARARE VALUES(911,802);

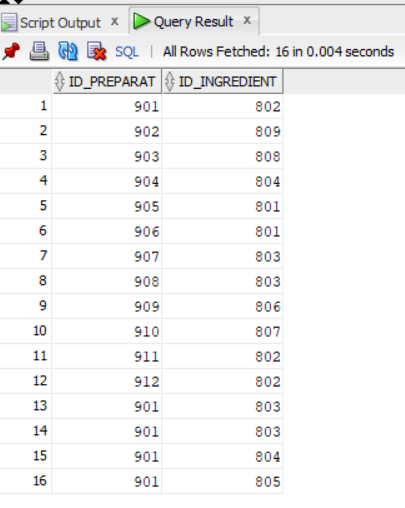
Insert into PREPARARE VALUES(912,802);

Insert into PREPARARE VALUES(901,803);

Insert into PREPARARE VALUES(901,803);

Insert into PREPARARE VALUES(901,804);

Insert into PREPARARE VALUES(901,805);



**6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.**

**Sa se afiseze angajatii unui restaurant al carui id este dat, salariul mediu pe toate restaurantele, salariul mediu pe restaurantul specificat, si daca angajatii din restaurantul specificat au salariu sub/peste medie.**

CREATE OR REPLACE PROCEDURE EX\_6(id RESTAURANT.ID\_RESTAURANT%type)

IS

TYPE Tablou\_Angajati is TABLE OF ANGAJATI%ROWTYPE index by pls\_integer;

TYPE tablou\_imbricat is TABLE OF ANGAJATI%ROWTYPE;

TYPE vector is varray(100) of ANGAJATI%rowtype;

toti\_ang tablou\_angajati;

tt vector;

ts tablou\_imbricat;

medie\_total int;

medie\_t int;

BEGIN

SELECT \* BULK COLLECT INTO toti\_ang from ANGAJATI;

DBMS\_OUTPUT.PUT\_LINE('TOTI ANGAJATII:');

DBMS\_OUTPUT.PUT\_LINE('---------------');

FOR i in 1..toti\_ang.COUNT loop

DBMS\_OUTPUT.PUT\_LINE(toti\_ang(i).nume\_angajat || ' --salariu : ' || toti\_ang(i).salariu\_angajat );

end loop;

select round(avg(salariu\_angajat))

into medie\_total

from angajati;

select round(avg(salariu\_angajat))

into medie\_t

from angajati where id\_restaurant=id;

DBMS\_OUTPUT.PUT\_LINE('SALARIU GENERAL MEDIU: ' || medie\_total);

DBMS\_OUTPUT.PUT\_LINE('SALARIU PE RESTAURANTUL ' || id || ' : ' || medie\_t);

SELECT \* BULK COLLECT INTO tt FROM ANGAJATI WHERE ID\_RESTAURANT= id AND SALARIU\_ANGAJAT > medie\_total;

SELECT \* BULK COLLECT INTO ts FROM ANGAJATI WHERE ID\_RESTAURANT= id AND SALARIU\_ANGAJAT < medie\_total;

FOR i in 1..tt.COUNT loop

DBMS\_OUTPUT.PUT\_LINE('Peste Medie: ' || tt(i).nume\_angajat || ' --salariu : ' || tt(i).salariu\_angajat );

end loop;

FOR i in 1..ts.COUNT loop

DBMS\_OUTPUT.PUT\_LINE('Sub Medie: ' || ts(i).nume\_angajat || ' --salariu : ' || ts(i).salariu\_angajat );

end loop;

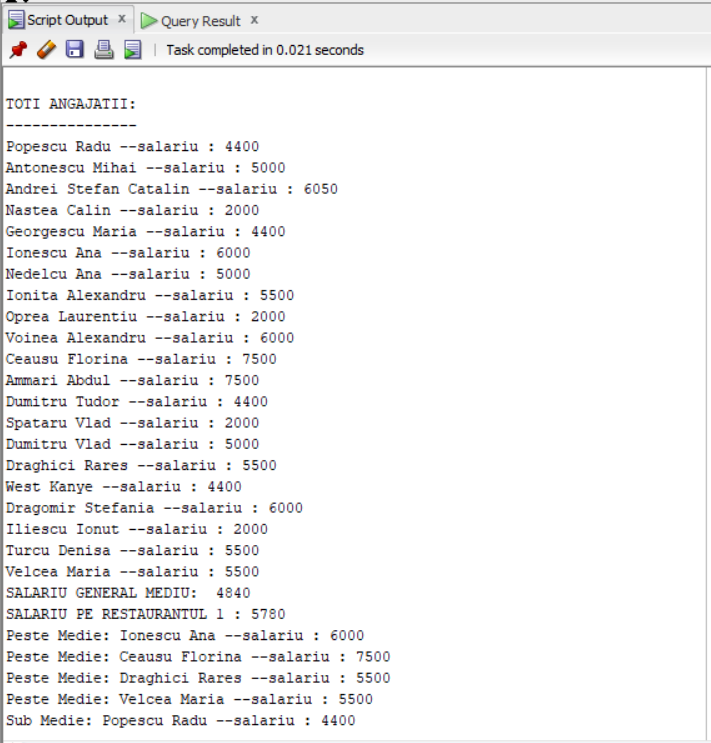
END;

/

BEGIN

EX\_6(1);

END;



**7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.**

**Sa se afiseze angajatii unui restaurant al carui id este dat , apoi sa se afiseze staff-ul restaurantului.Se cere si afisarea chefilor din restaurantul respectiv a caror specialitate este data .**

CREATE OR REPLACE PROCEDURE ex7 (id\_rest IN RESTAURANT.ID\_RESTAURANT%type,p\_spec IN VARCHAR2)

AS

cursor c\_ang(id\_rest IN RESTAURANT.ID\_RESTAURANT%type) is

select \* from ANGAJATI a

where a.id\_restaurant=id\_rest;

cursor c\_chefi is

select \* from chef ch

join angajati a on ch.id\_angajat=a.id\_angajat

where a.id\_restaurant=id\_rest AND ch.specializare=p\_spec;

v\_angajati c\_ang%ROWTYPE;

v\_chefi c\_chefi%ROWTYPE;

cursor c\_staffr(id\_rest IN RESTAURANT.ID\_RESTAURANT%TYPE) IS

select \* from STAFF\_RESTAURANT s JOIN ANGAJATI a ON s.ID\_ANGAJAT=a.id\_angajat

where a.id\_restaurant=id\_rest;

v\_staffr c\_staffr%ROWTYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(' ANGAJATI RESTAURANTULUI ' || id\_rest);

DBMS\_OUTPUT.PUT\_LINE('--------------------------');

OPEN c\_ang(id\_rest);

LOOP

FETCH c\_ang INTO v\_angajati;

EXIT WHEN c\_ang%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_angajati.NUME\_ANGAJAT);

END LOOP;

CLOSE c\_ang;

DBMS\_OUTPUT.PUT\_LINE('--------------------------');

DBMS\_OUTPUT.PUT\_LINE(' STAFF-UL RESTAURANTULUI ');

DBMS\_OUTPUT.PUT\_LINE('--------------------------');

OPEN c\_staffr(id\_rest);

LOOP

FETCH c\_staffr INTO v\_staffr;

EXIT WHEN c\_staffr%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('STAFF : '||v\_staffr.NUME\_ANGAJAT ||' Salariu: ' ||v\_staffr.SALARIU\_ANGAJAT );

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('--------------------------');

DBMS\_OUTPUT.PUT\_LINE('CHEFII RESTAURANTULUI ');

DBMS\_OUTPUT.PUT\_LINE('--------------------------');

OPEN c\_chefi;

LOOP

FETCH c\_chefi INTO v\_chefi;

EXIT WHEN c\_chefi%NOTFOUND;

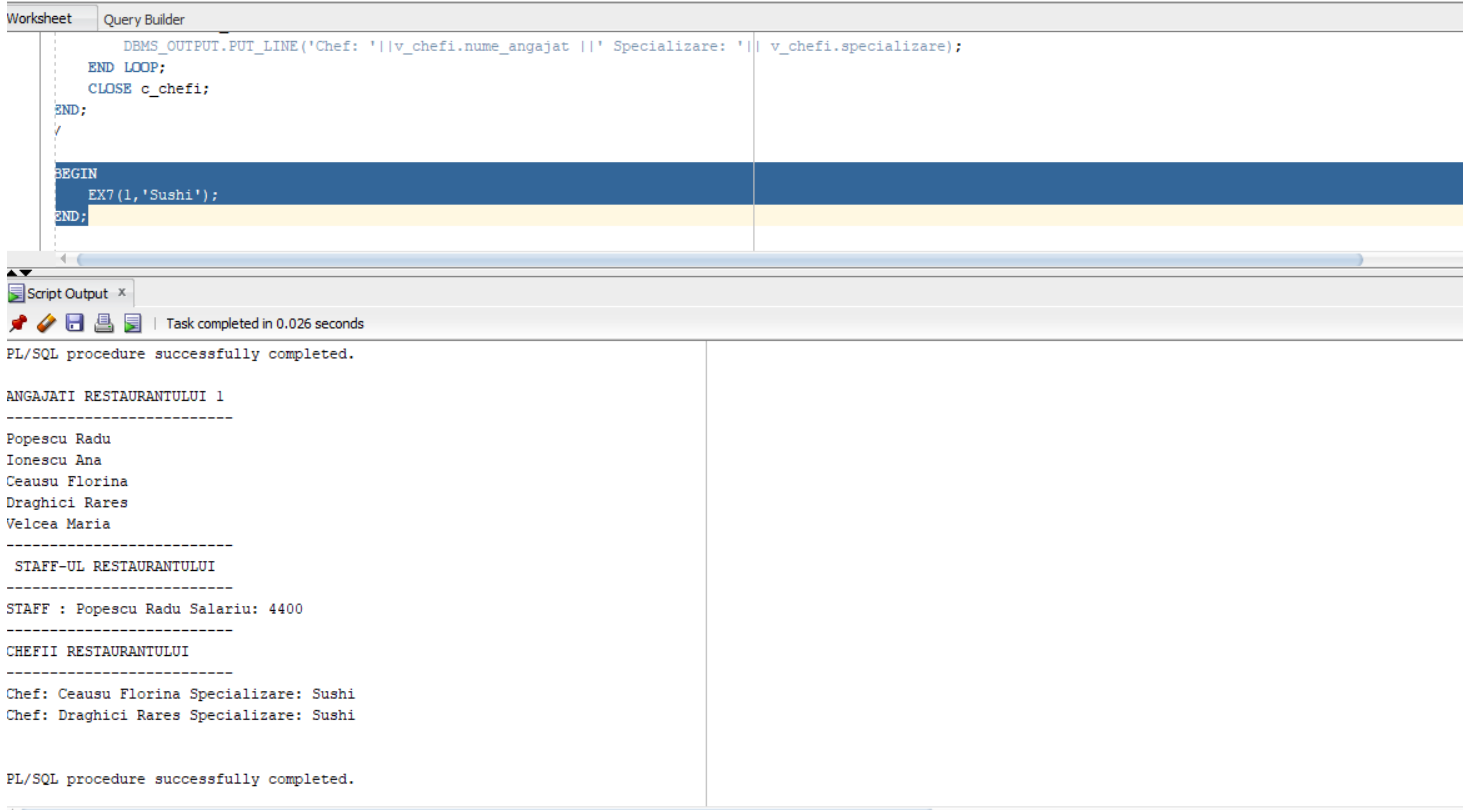
DBMS\_OUTPUT.PUT\_LINE('Chef: '||v\_chefi.nume\_angajat ||' Specializare: '|| v\_chefi.specializare);

END LOOP;

CLOSE c\_chefi;

END;

/



**8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite.**

**Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.**

**Pentru un membru din staff al carui id este dat, sa se afiseze suma totala din comenzile efectuate de acesta. Membrii din staff care nu au cel putin 20 de ore de experinta sau cel putin 2 comenzi efectuate nu sunt luati in considerare.**

CREATE OR REPLACE FUNCTION EX8(p\_id\_staffr IN NUMBER) RETURN VARCHAR2 AS

v\_suma NUMBER;

v\_total NUMBER;

v\_cnt NUMBER;

v\_cursor SYS\_REFCURSOR;

v\_ore\_experienta NUMBER;

ID\_INVALID EXCEPTION;

PREA\_PUTINE\_COMENZI EXCEPTION;

EXPERIENTA\_INSUFICIENTA EXCEPTION;

BEGIN

IF p\_id\_staffr < 501 OR p\_id\_staffr > 506 THEN

RAISE ID\_INVALID;

END IF;

SELECT ORE\_EXPERIENTA INTO v\_ore\_experienta FROM STAFF\_RESTAURANT

WHERE ID\_STAFFR=p\_id\_staffr;

IF v\_ore\_experienta < 20 THEN

RAISE EXPERIENTA\_INSUFICIENTA;

END IF;

OPEN v\_cursor FOR

SELECT p.SUMA\_TOTALA FROM PLATA p

JOIN COMANDA c ON p.ID\_COMANDA = c.ID\_COMANDA

JOIN STAFF\_RESTAURANT s ON c.ID\_STAFFR = s.ID\_STAFFR

WHERE s.ID\_STAFFR=p\_id\_staffr;

v\_suma :=0;

v\_total:=0;

v\_cnt:=0;

LOOP

FETCH v\_cursor INTO v\_suma;

EXIT WHEN v\_cursor%NOTFOUND;

v\_total:= v\_total + v\_suma;

v\_cnt:=v\_cnt+1;

END LOOP;

CLOSE v\_cursor;

IF v\_cnt < 3 THEN

RAISE PREA\_PUTINE\_COMENZI;

END IF;

RETURN v\_total;

EXCEPTION

WHEN ID\_INVALID THEN

DBMS\_OUTPUT.PUT\_LINE('ID-ul introdus este invalid!');

RETURN NULL;

WHEN PREA\_PUTINE\_COMENZI THEN

DBMS\_OUTPUT.PUT\_LINE('Membrul staff nu a efectuat suficiente comenzi!');

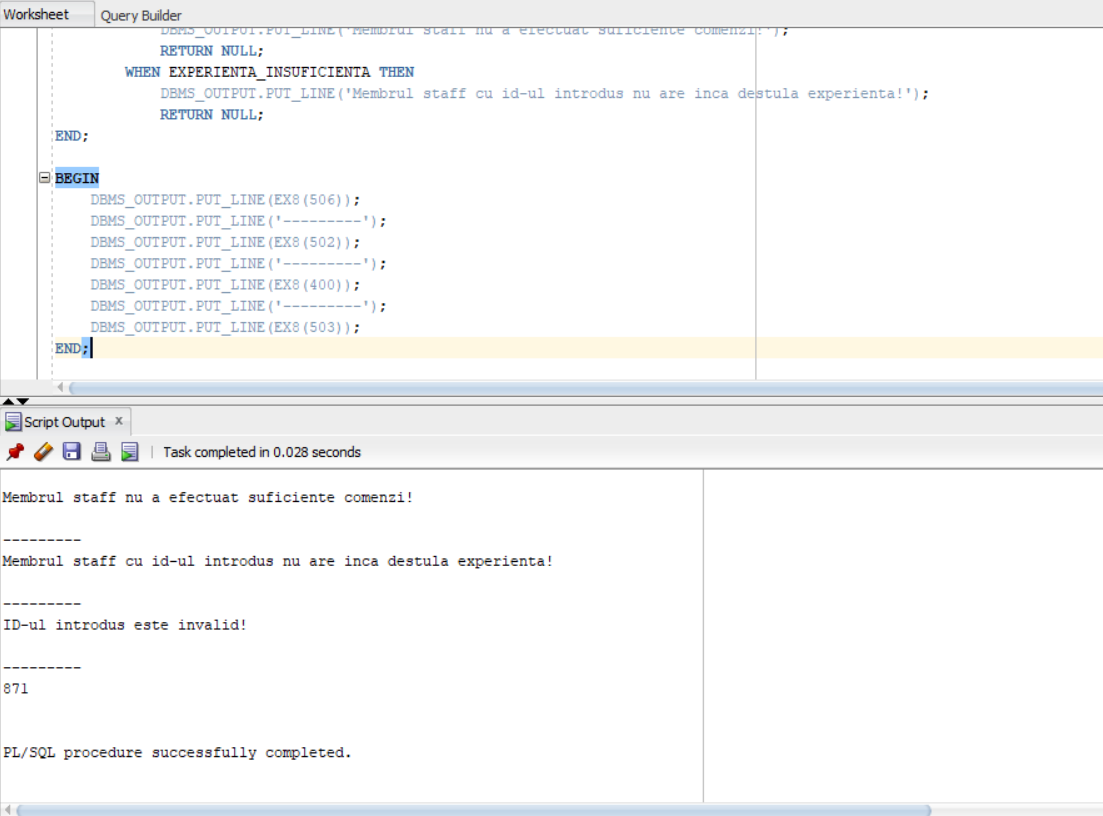
RETURN NULL;

WHEN EXPERIENTA\_INSUFICIENTA THEN

DBMS\_OUTPUT.PUT\_LINE('Membrul staff cu id-ul introdus nu are inca destula experienta!');

RETURN NULL;

END;



**9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

**Sa se afiseze pentru un client al carui nume ii este dat, cel mai scump preparat comandat(nume+pret) impreuna cu id-ul platii respective .**

CREATE OR REPLACE PROCEDURE ex9(p\_nume\_client IN VARCHAR2) AS

v\_nume\_preparat PREPARAT.NUME\_PREPARAT%TYPE;

v\_pret\_preparat PREPARAT.PRET%TYPE;

v\_id\_plata PLATA.ID\_PLATA%TYPE;

v\_nume\_client CLIENT.NUME\_CLIENT%TYPE;

v\_init CLIENT.NUME\_CLIENT%TYPE := 'gol';

final\_nume\_preparat PREPARAT.NUME\_PREPARAT%TYPE;

final\_pret\_preparat PREPARAT.PRET%TYPE := 0;

final\_id\_plata PLATA.ID\_PLATA%TYPE;

final\_nume\_client CLIENT.NUME\_CLIENT%TYPE;

CURSOR c\_items IS

SELECT pr.NUME\_PREPARAT,pr.PRET,p.ID\_PLATA,cl.NUME\_CLIENT

FROM CLIENT cl

JOIN COMANDA co ON co.id\_client = cl.id\_client

JOIN PLATA p ON p.id\_comanda=co.id\_comanda

JOIN ITEM\_COMANDA i ON i.id\_comanda=co.id\_comanda

JOIN PREPARAT pr ON pr.ID\_PREPARAT=i.id\_preparat

WHERE UPPER(cl.NUME\_CLIENT) LIKE '%' || UPPER(p\_nume\_client) || '%';

INPUT\_INVALID EXCEPTION;

WRONG\_TYPE EXCEPTION;

BEGIN

IF regexp\_like (p\_nume\_client,'[0-9]') THEN

RAISE WRONG\_TYPE;

END IF;

IF p\_nume\_client IS NULL THEN

RAISE INPUT\_INVALID;

END IF;

OPEN c\_items;

LOOP

FETCH c\_items INTO v\_nume\_preparat,v\_pret\_preparat,v\_id\_plata,v\_nume\_client;

EXIT WHEN c\_items%NOTFOUND;

if v\_init !=v\_nume\_client and v\_init!='gol' then

raise TOO\_MANY\_ROWS;

END IF;

v\_init:= v\_nume\_client;

IF(v\_pret\_preparat > final\_pret\_preparat) THEN

final\_nume\_preparat := v\_nume\_preparat;

final\_pret\_preparat :=v\_pret\_preparat;

final\_id\_plata :=v\_id\_plata;

final\_nume\_client :=v\_nume\_client;

END IF;

END LOOP;

IF v\_init = 'gol' then

RAISE NO\_DATA\_FOUND;

END IF;

DBMS\_OUTPUT.PUT\_LINE( ' Nume preparat si pret: '||final\_nume\_preparat || ' ' || final\_pret\_preparat|| ' Id plata: ' || final\_id\_plata || ' Nume Client: ' ||final\_nume\_client);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Exista mai multi clienti cu acest nume!');

RETURN;

WHEN WRONG\_TYPE THEN

DBMS\_OUTPUT.PUT\_LINE('Numele clientului nu poate contine cifre!');

RETURN;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun client cu numele '|| p\_nume\_client );

RETURN;

WHEN INPUT\_INVALID THEN

DBMS\_OUTPUT.PUT\_LINE('Numele clientului nu poate sa fie null!' );

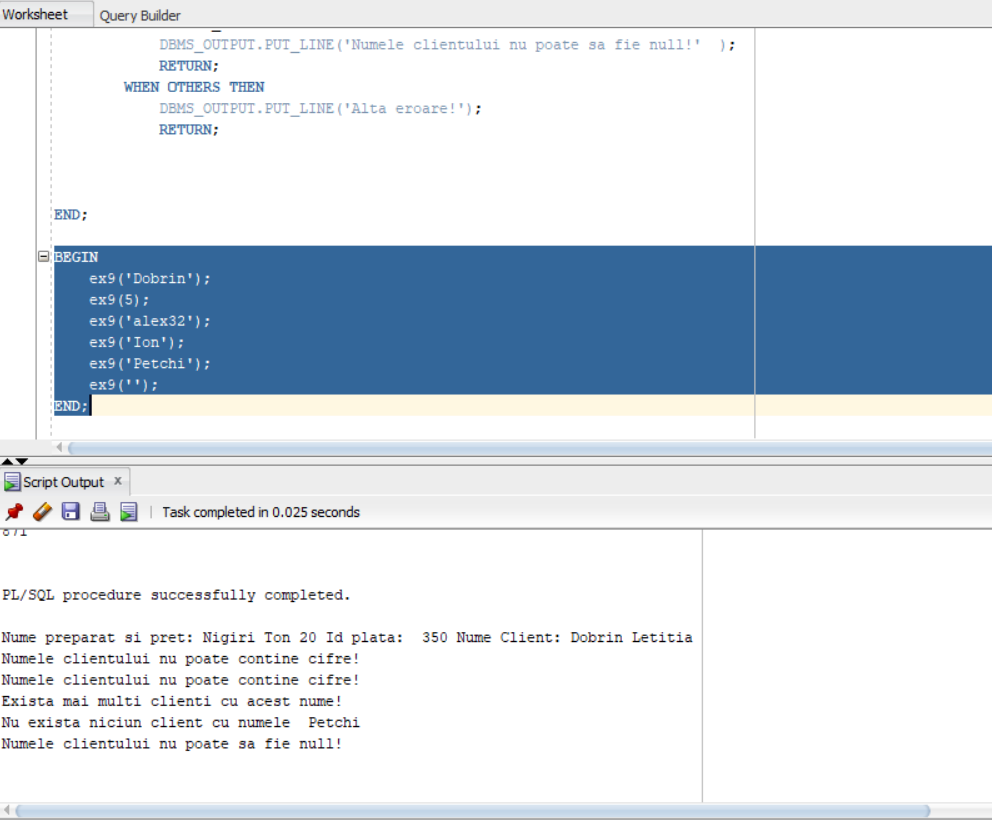
RETURN;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Alta eroare!');

RETURN;

END;



**10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.**

**Creati un trigger care interzice angajarea personalului in luna ianuarie.**

CREATE OR REPLACE TRIGGER ex10

BEFORE INSERT OR UPDATE ON ANGAJATI

BEGIN

if(to\_char(sysdate,'DD-MM')>'01-01' and to\_char(sysdate,'DD-MM')<'31-01')

then raise\_application\_error(-20001,'Nu se poate angaja personal in luna ianuarie');

end if;

end;

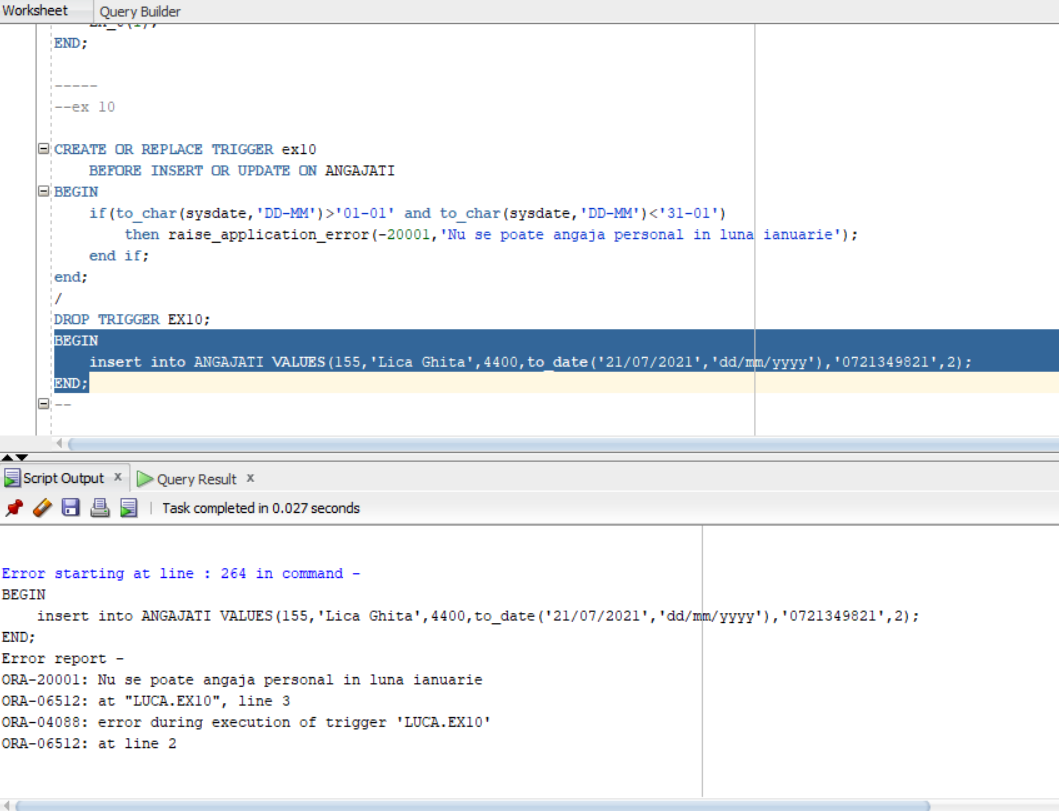
/

DROP TRIGGER EX10;

BEGIN

insert into ANGAJATI VALUES(155,'Lica Ghita',4400,to\_date('21/07/2021','dd/mm/yyyy'),'0721349821',2);

END;



**11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.**

**Creati un trigger care interzice modificarea datelor de nastere a angajatilor**

CREATE OR REPLACE TRIGGER ex11

BEFORE UPDATE OF DATA\_ANGAJAT ON ANGAJATI

FOR EACH ROW

WHEN (NEW.DATA\_ANGAJAT!= OLD.DATA\_ANGAJAT)

BEGIN

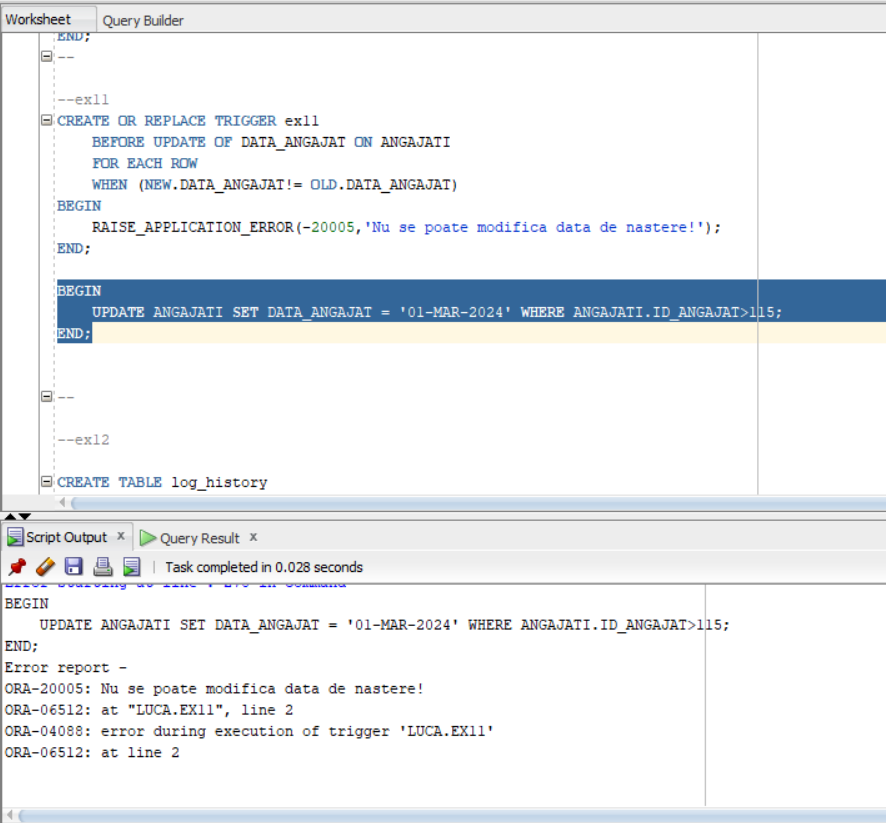
RAISE\_APPLICATION\_ERROR(-20005,'Nu se poate modifica data de nastere!');

END;

BEGIN

UPDATE ANGAJATI SET DATA\_ANGAJAT = '01-MAR-2024' WHERE ANGAJATI.ID\_ANGAJAT>115;

END;



**12. Definiți un trigger de tip LDD. Declanșați trigger-ul.**

**Creati un trigger care sa permita modificarea schemei doar de utilizatorul Luca. Sa se salveze modificarile facute asupra schemei.**

CREATE TABLE log\_history

(

username VARCHAR2(20),

log\_date DATE,

event varchar2(100)

);

CREATE OR REPLACE TRIGGER ex12

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

IF USER != 'Paul' THEN

RAISE\_APPLICATION\_ERROR(-20910,'Numai administratorul bazei de date poate realiza operatia');

END IF;

INSERT INTO log\_history VALUES(SYS.LOGIN\_USER,SYSDATE,SYS.SYSEVENT);

END;

/

DROP TRIGGER EX12;

CREATE TABLE testt(id NUMBER);

SELECT \* FROM log\_history;

